Autonomous pathfinding for planetary rover by implementing A* algorithm on an aerial map processed using matlab image processing tool.

Dhyanchand Jani
Department of Mechanical Engineering,
National Institute of Technology, Rourkela

Autonomous pathfinding for planetary rover by implementing A* algorithm on an aerial map processed using matlab image processing tool.

Report submitted for fulfilment of the requirements of ME 591(Research Project)

By

Dhyanchand Jani

(711ME4074)

(10^{th} semester, Department of Mechanical Engineering, NIT Rourkela)

Under The Guidance of

Dr.Dayal Ramakrushna Parhi

DEPARTMENT OF MECHANICAL ENGINEERING

NATIONAL INSTITUTE OF TECHNOLOGY, ROURKELA

March 2016

Department of Mechanical Engineering
National Institute of Technology, Rourkela

Prof. Dayal Ramakrushna Parhi

Professor

May 25, 2016

Supervisor's Certificate

This is to certify that the work presented in this dissertation entitled Autonomous path finding for a planetary rover by implementing A* algorithm on an aerial map processed using MATLAB image processing tool" by “Dhyanchand Jani", Roll Number 711ME4075, is a record of original research carried out by him under my supervision and guidance in partial fulfilment of the requirements of the degree of *Master of Technology in Mechatronics and Automation*. Neither this dissertation nor any part of it has been submitted for any degree or diploma to any institute or university in India or abroad.

Prof. Dayal Ramakrushna Parhi

Dedicated to

My loving parents & supportive friends

Declaration of Originality

I, Dhyanchand Jani, having Roll Number 711ME4075, hereby declare that this dissertation entitled "**Autonomous Pathfinding for a planetary rover by implementing A* algorithm on an aerial map processed using matlab image processing tool** " represents my original work carried out as a Dual degree student of NIT Rourkela and, to the best of my knowledge, it contains no material previously published or written by another person, nor any material presented for the award of any other degree or diploma of NIT Rourkela or any other institution. Any contribution made to this research by others, with whom I have worked at NIT Rourkela or elsewhere, is explicitly acknowledged in the dissertation. Works of other authors cited in this dissertation have been duly acknowledged under the section "References". I have also submitted my original research records to the scrutiny committee for evaluation of my dissertation.

I am fully aware that in case of any non-compliance detected in future, the Senate of NIT Rourkela may withdraw the degree awarded to me on the basis of the present dissertation.

Date: Dhyanchand Jani

NIT Rourkela 711ME4075

Acknowledgement

I wish to express my sincerest gratitude towards my project guide Prof. Dayal Ramakrushna Parhi whose support and encouragement during the work which helped me to achieve the objective of the project. His continual supervision and enthusiasm has propelled me and my fellow project colleague. Hence, the research work became smooth and enjoyable for me.

I am deeply indebted to my fellow colleague Sourav panda for assisting me throughout the duration of entire research work.

At last, I also take this opportunity to extend my deep appreciation to family and friends, for providing me all their moral support during the crucial times of my project.

Place: Rourkela

Date:

Dhyanchand Jani
Department of Mechanical Engineering
National Institute of Technology Rourkela

Abstract

Human curiosity to discover new things and exploring unknown regions, have continually to development of robots, which became a powerful tools for accessing dangerous environments or exploring regions too distant for human. Previous robot technology functioned under continues human supervision, limiting the robot to confined area and pre-programmed task. However, as exploration moved to regions where communication is ineffective or unviable, robots were used to carry out complex tasks without human supervision. To empower such capacities, robots are being upgraded by advances extending from new sensor improvement to automated mission planning software, circulated automated control, and more proficient power systems. With the advancement of autonomy science robotics technology developed and the robots became more and more capable of operating multi task, under minimal human supervision.

In this project work we aim at designing an ONS (Offline Navigation System) system for the planetary rover which will use aerial map taken from satellite and pre-process into a grid map which is then will be used by the rover to travel from one place to another place and completing its mission. The aerial map is processed using Matlab image processing tool to convert into a grid map and search for shortest route is implemented using A* algorithm. The shortest route result is then converted into microcontroller signal to move the rover. With this system the rovers will have the ability to predict the best possible path even if the communication to the satellite is broken.

Keywords: Offline navigation, Path planning, image processing, A* algorithm

Contents

List of figures

List of Tables

Chapter 1 – INTRODUCTION

Overview

Background

Motivation & Objective

Literature Review

Thesis Organisation

1.1 Overview

Since the 1970s, research on planetary rovers has been conducted at various times at JPL. In the 1980s, it was planned to use a highly capable Mars rover to traverse long distances, perhaps aided by high-resolution (roughly one-meter) stereo pictures taken from orbit, so that only occasional commands from Earth would be needed (Randolph, 1986; Mankins, 1987; Wilcox and Gennery, 1987; Gennery, 1989). However, because of the propagation delay from Earth, considerable autonomy in a Mars rover is highly desirable, especially if it must drive far between sample sites. Also, because of the difficulty in communication, autonomy may be needed for a vehicle on the far side of the Moon. This paper describes some techniques that will be useful in path planning by such vehicles.

1.2 Background

The drive to explore is a human instinct that has never changed. To understand the world and the universe we live, space exploration programmers have been initiated to distant and unknown environments with huge capital investments. As we explore, the technical challenge and the cost of keeping the rover in their mission has led to need of an efficient path planning system which will rely less on satellite and focus more on providing the autonomy to the rover itself.

The autonomous path planning is an essential function of every exploration rover to avoid obstacles, which can be divided into two categories:

1.2.1 Map generation

The first step of every path planning is to get information about its environment and create a grid map with respect to which the rover will travel from one point to another point. These are generally a set of nodes derived by segmentation of the actual map.

1.2.2 Search algorithm

The search algorithm are a set of codes run on the map generated to find the efficient path for the rovers. The path traversed can be perfect path for shortest route or least cost path to make traversing cost as low as possible.

The A* algorithm is computer algorithm which is used widely in pathfinding and graph traversal. It combine the formal approach like Dijkstra’s and the heuristic approach like Greedy-Best-First-Search to find an efficiently traversable path between multiple nodes.

1.3 Motivation

The surface terrain of the other planets are covered with hilly terrain, craters, small pit and canyon enough to engulf the rover. Because of this challenging terrain, the rover need to evade these obstacle prior to the contact from the local sensor. If a rover only works using local sensor, then it will not know about the global obstacle present ahead which needs to be avoided to follow a least power consumption path or even evade mobility hazards, which includes vehicle rollover, immobilizing wheel slippage, or collision with obstacle rocks. Therefore, the planetary rover needs to navigate itself by sensing the global environment as well as planning a feasible path for rough terrain traverse.

Since, GPS is not available on Mars, path estimating for the rover using traditional odometry method is likely to produce some errors. Also these odometry path estimation are done million miles away on earth and sending request to earth & getting the commands from earth is mins long time. In order to overcome these problems we need to provide the rover the tool to find its own path using the aerial map of the satellite.

1.4 Objective

The rovers are remotely operated vehicle used to explore the surface of the other planets to gather information about the possibilities of life and sustainability of life if a colony is established. To carry out these humankinds greatest discovery mission they need to complete their task without getting in trouble. The movement of these rovers on other planet, million miles away, are operated form the earth. It takes mins of time to send and receive one signal which leaves the operator renderless for that duration of time. Also as a huge capital is invested in these mission it is very important for the rover to complete its mission. And to move the rover from one place to another, an autonomous navigation system is a must necessary for these rovers.

Some of the unavoidable problems of a planetary rovers which directly or indirectly affects the rover operating life if an autonomous system is not there:

- Signal transmission and receiving takes long time.
- Fixed power holding capacity makes them incapable of going long distances.
- If there is a communication failure then the rover will stop working.

The objective of this paper is to provide the ability to deduce their own path even if the communication from the satellite is not available. The rover will have the aerial map of the region using, which it will travel to its destination and complete its mission and also safely return to the base. These grid map can be stored to be further used in case of further mission on the same area.

1.5 Thesis organisation

Table 1.1 Overview of thesis according to chapters

Chapter	**Content**	**Remarks**
Chapter 1	Introduction	Has back background, motivation and objectives of the project
Chapter 2	Literature survey	Contains literature survey in relevance to the project
Chapter 3	Methodology	Theoretical aspects of the methods used image processing, A* star search algorithm
Chapter 4	Experimental setup	Details of the parts of the system
Chapter 5	Code implementation and results	Working of the system, algorithm and results of the simulation.
Chapter 6	Conclusions and future scope of work	Conclusions of the experimental setup and Future scope of work discussed in the project

Chapter 2 – LITERATURE SURVEY

The path planning for a space rover is the central topic of interest citing several research application currently ongoing in this field of interest. While there are numerous number of path finding methods available, it is must necessity that the path planning method is well suited for the target environment. In this survey we study different types of path planning methods and to derive a suitable simple method.

In 2015, at jet propulsion laboratory**, Masahiro.Ono et al.** [1] worked on risk avoidance system for efficient path planning. As identifying and avoiding surface terrain hazards is a crucial for a space exploration rover, they tried to focus their research on identifying these problems. They developed a ground based mars rover operation tool that identify the pointy rocks, small cracks/pits on the terrain, evaluating their risks, and finding out safe through them. The system helps in avoiding the high cost paths, by reducing the decision making load on the operator, by preventing human error, most importantly, by significantly reducing the risk of loss of rover. The risk-aware system is based on two important technologies. The first method is an image processing terrain classification system that is capable of identifying potential hazards, such as bold rocks and soft terrains, from images. The second is a path planning system based on rapidly-exploring random graph (RRG) and the A* search algorithms, which avoids the rocky terrain efficiently controls the wheel displacement. The images taken by rover are first processed through the terrain identifying system then it is processed for path of least resistance.

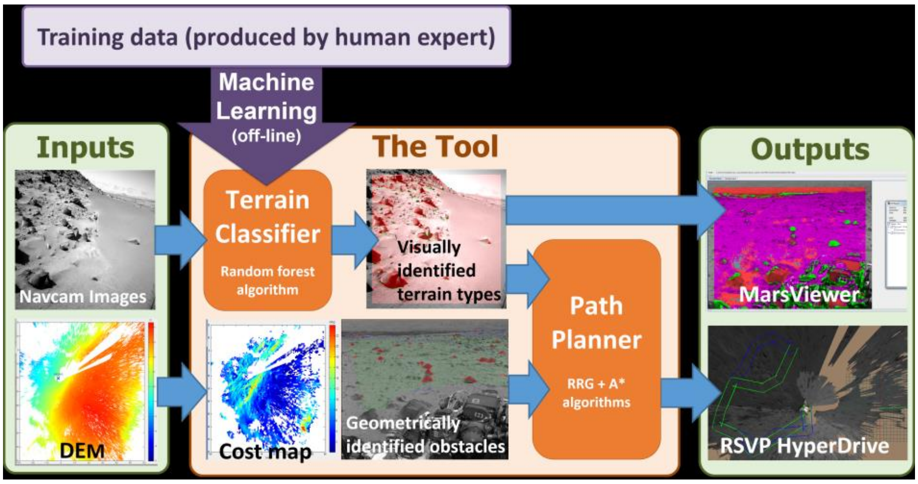

Figure 2.1: Working of the risk avoidance system.

In the paper, published by **Fredrich Fraundorfer et al. [2]**, in 2012, he tried to create a system that uses camera as its main sensor. The project was on autonomous vision based quadrotor MAV system which takes picture of its unknown environments, makes maps of them and then again explores unknown environments. All the necessary equipment needed for the system was on board, the autonomous mapping and exploration system also. Using only a stereo camera placed at the front of, the quadrotor achieves the abilities to create a Vector Field Histogram+ (VFH+) algorithm for local navigation, and unknown frontier exploration algorithm. In addition to it, a bug algorithm was implemented for autonomous wall-following which is substitute for the unknown frontier exploration where the frontier-based exploration under-performs. With the discovery of unknown regions it side by side builds a 3D global map on-board the MAV. This map is then used by the VFH+ and frontier-based exploration in dense environments, and the Bug algorithm of wall-following is used in case of sparse environments. During the exploration phase, images from the front camera are transmitted to the ground station via Wi-Fi. These images are then transferred to the ground station which concates these into a large scale map. SLAM is carried out with pose-graph optimization and loop closure detection using a vocabulary tree. The robustness of the pose estimation is improved by fusing optical flow and visual odometry. Another camera is provided at the bottom looking downward which provides the top view of the images. The visual odometry is only measured using the front camera only.

Yao-hong Qu et al. [3] proposed flight path planning for a UAV based on heuristic search and genetic algorithm. He divided his research into two parts: one is optimal path planning without considering the computation cost and real-time suboptimal path planning. The objective of the two methods was to find a suitable path planning method for UAV. The first method is an offline path planning in which the path is first calculated using the genetic algorithm. Because despite of all safety arrangements or tight algorithm there will always be some shortcomings. In order to tackle those situation genetic algorithm is used. At first a set of threats ’Delaunay triangle net based are constructed using the nearest neighbourhood principle method. Then designate each point (position of a threat) on the left of the path a index 0 and on the right 1 and only designate the points of line which is not passed the same symbol. Thus based on the intersection of the Delaunay triangle, a kind of encoding is designed on the principle of “Left0, Right 1”. At last a global path planning is obtained using genetic algorithm and potential field technology. The other one is based on heuristic A* approach method. In this method, by using the A-star algorithm we search the threat and then a shortest path is obtained, which is consists of the lines on the Voronoi diagram, and then by taking into consideration the turning constraint of the UAV the path can be smoothened using geometry method.

Charles W. Warren et al. [4] in 1993, published a paper on path planning for a robot surrounded by cluttered obstacle using a modified A* method for searching through the free space. The A* method is the simplest heuristic search algorithm and also the most commonly used path planning algorithm, but when comes to doing searching in large number of cells it is very slow. To improvise the search algorithm he used trial vectors for spanning across several cells. In this way a good obstacle mapping is obtained with a loose search is performed on the grid. The result for of this method are that it increased the path planning /searching by several milliseconds compared to the several minute of traditional A* algorithm.

Li Maohai et al. [5] conducted an experiment to find out a robust topological navigation system for an omnidirectional robot. The experiments consists of two different approach: Offline navigation and online navigation. In offline strategy the robot uses performs a omnidirectional motion with the help of which it takes picture of it’s surrounding through the Omni directional camera. Form all those taken, it creates a topological map based on a probabilistic technique and loop closure detection algorithm. Every topological node has a set of omnidirectional images characterized by geometrical data. Whenever a target is selected. It concates all the topological images to create a map. Thus finds its way towards the target. In the online navigation strategy the robot use classical 5 point pose estimation algorithm to find out location. The robot then is follows a vision based control law constructed especially for the omnidirectional camera.

Korbinian Schmid et al. [6] in 2013, developed an autonomous flying system which has a fully functional onboard system capable of all image processing and the path planning. The flying system is a quadrotor with a stereo camera on it, it also has an IMU, tow processor and an FPGA to process the image taken by the stereo cameras. Stereo images are processed on the FPGA by the Semi-Global Matching algorithm. The complete frame is fused with IMU based stereo odometry. The system gathers data creates a 3D map.If an operator selects two points in the map, then the quadrotor autonomously plans of obstacle free and fly's along it.

Thomas M. Howard et al. [7] His research work on continues planetary rover navigation through FPGA stereo and visual odometry focuses on finding the least resistances path for the rover. The main key concerns for a rover with low computational power is moving safely through the environment while consuming a minimal amount of computational resources, energy and time. Three most computational intensive process in this autonomous path planning era is perception, pose estimation and motion planning. Even if to move a distance of 1meter the rover has to continually process it environment then find the rate of change of pose and accordingly generate the next mobility maneuver.

This paper describes improvements in the energy efficiency and speed of planetary rover autonomous mobility system accomplished by transferring the process that are done by CPU to FPGA coprocessor. Perception algorithms in general are well suited to FPGA implementations because much of processing is naturally parallelizable. In this paper we tend to implement the new idea of implementing stereo visual odometry processing onto the FPGA processor for better mobility and increased performance. The FPGA stereo implementation used for rectification and high performance between disparity image, filtration and rectification. The enhanced visual odometry component uses FPGA implementation. The FPGA implementation of the stereo and visual odometry functionality have resulted in an increase in the performance approximately three time compared to the MER-class avionics.

Stereo vision navigation is a challenging area in Robotic research. An efficient swarm robotic system was proposed by **Aritra Sarkar et al. [8]** for mapping of unknown regions by use stereo vision. They also included a cloud computing system in the communication module via private cloud over OLSR. The system uses real visual odometry technique, disparity mapping, and cloud computing and network hazard recovery algorithm for efficient terrain mapping. The map is stored in an octree data structure. The system is designed keeping in perspective planetary exploration mission, however it is widely scalable and can be modified for various civilian and military purposes.

Genya Ishigami et al. [9] in this paper a path planning system and a navigation framework for a planetary exploration rover using laser range finder device is modelled. The laser range finder compute accurately all the obstacle present in the surrounding and provides a good terrain mapping. Then the path planning algorithm creates a suitable path based on cost function consisting of terrain inclination, terrain toughness and terrain length. A set of navigation commands for the rover is then computed from the generated path.
After the path is generated the rover follows the commands to reach the desired goal.

Jerome Barraquand et al. [10] proposed a new approach towards robot path planning which consists of building a graphical map by joining the local minima of a potential field defined in robot's configuration. The robot continually search for the local minima and using those graph it created till it find the target. Opposite to the global mapping, this technique uses doesn't requires any expensive computation step before the search for a path can actually start. On the other hand, it searches a graph that is usually much smaller than the graph searched by the so-called "local "methods. This technique can be described in two ways. 1) Constructing "good "potential fields and 2) efficiently escape their local minima.

Ron kimmel et al. [11] on 2001, published a paper discussing an optimal algorithm for path planning by the reconstruction of 3D terrains from shaded image. The algorithm used involves single shading image problem. Here the shaded image is used a input image and the height of the reconstructed surface weighted distance. A machine learning numerical algorithm based on Sethian's fast marching method is used to reconstruct the terrain surface from the shaded image. The 3D surface obtained is a dense solution of an Eikonal equation for the vertical light source case. For the oblique light source case, the reconstructed surface is a equivalent solution to a different partial differential equation. A modification of the fast marching method gives a numerically stable, optimal solution, and practically fast algorithm for the classical shape from shading problem. In another set of problem the fast marching method coupled backed up a back tracking algorithm via gradient descent along the reconstructed surface to solve the path planning problem in robot navigation.

Youcef Mezouar et al. [12] To make an image based path planning one needs a vision feedback system for efficient and robust path planning. Sometimes the position of the rover is very far from the target position that, so in order to reach the target the robot have to pass through numerous obstacles. And for an image based robot to achieve those things a continues vision feedback loop. Even after all the classical based on the zero error function computed the current measurement and a constant. It is therefore required to create a link between the image based control and the path planning in vision space. Constraints, such that the object should remain in the field of view of camera or the restriction in the joint part can be taken into account at the task planning level.

Furthermore, with the help of this method it is ensured that the robot measurements always remains within the restricted parameters and a control via image-based ensures that the robustness is maintained with respect to modelling errors. The given method is based on the potential field approach and its application to check whether the object shape and dimensions are known or not, and whether the calibration parameters of the camera are well or badly estimated. Finally, real-time experimental results using an eye-in-hand robotic system are presented and confirm the validity of our approach.

In paper on visual odometry path estimation paper by **Peter Corke and Dennis Strelow et al.** [13] shows the comparison between the traditional robust optical flow method and visual odometry from a omnidirectional image sequence. In their conclusion they concluded that the visual odometry is a very accurate method of path planning even though it has very large computation expense which can be covered using the latest computing technology. Early path estimation has been typically limited to odometry based on proprioceptive measurements such as the integration of distance travelled and measurement of heading change. But with vision technology all this integration need not to be done in place of which some simple but long image processing problems will be carried out.

Chapter 3 – METHODOLOGY

Image processing

A star search algorithm

Serial transmission

3.1 Image processing: Map generation

Map generation is done using MATLAB image processing tools. The image processing tool box provides a set of standard algorithms, function, and apps for image processing, analysis, visualization, and algorithm development. Using this tool we can perform image segmentation, noise reduction, and geometric transformation.

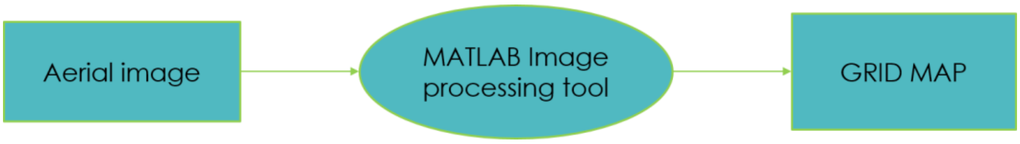

Figure 3.1: Aerial image to grid map conversion flow chart.

The image processing is done in following three stages:

1) Image acquisition
2) Rover detection
3) Grid map creation

3.1.1 Image acquisition

This operation tool contains function that connects the camera with the MATLAB environment. Then with the "videoinput" function it locates the port where camera is connected and establishes a connection. Using the "get", "set" function the properties of the video input is configured. After that the video streaming is started using "startvid" function from which snapshot of picture containing the rover and destination is taken and stored in memory.

3.1.2 Rover and Obstacle detection

The image is then processed for rover and obstacle detection. By using the object detection operation and noise filtration technique of MATLAB image processing the rover and obstacle are found. The width of the rover is found (in pixels) with respect to the size of the map for grid map creation.

3.1.3 Grid Map creation

By using the size of the rover as reference the entire image is then divided into square cells (each cell size = size of rover). Each cell is assigned a particular node number from top left corner as cell number 1 to bottom right cell as the last node of the image division. To keep the link between the actual map and the grid map, the grid map created is a 3 dimensional array with each node cell containing the x co-ordinate, y co-ordinate of the center pixel of the image area enclosed by the node cell in the image.

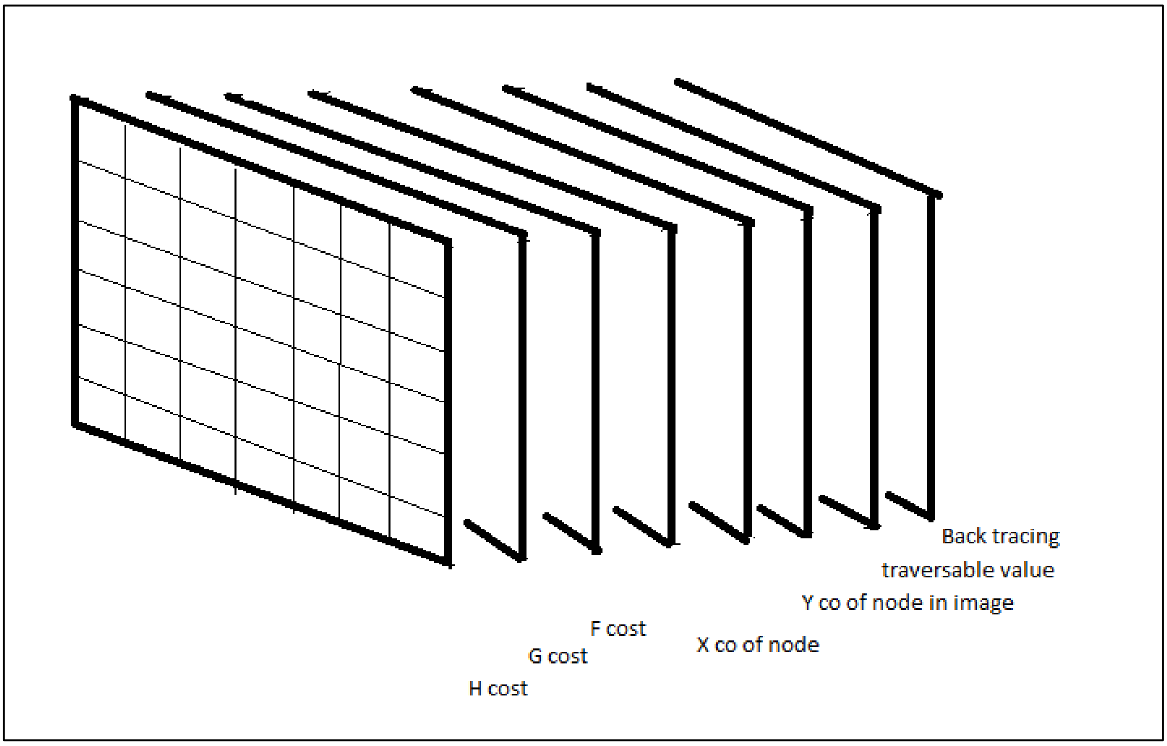

Figure 3.2: Theoretical image of the grid map.

The 3D grid map constructed here has 7 z planes. The 4th and 5th plane are the linkage planes, for relating the rover position in grid map with that of the actual map. The 1st, 2nd and 3rd are the cost detail plane to keep track of the movement cost of the rover. The 6 plane keeps track whether a node is traversable or not. The value for determining whether a node is traversable or not is decided after obstacle detection in the map is done. The node with the obstacle is marked 0 while the walkable one is valued 1. This traversable plane gets update whenever a new obstacle come along the path. The last plane (7th plane) is the tracing plane for keeping track of the steps of rover movement. The figure xx shows an example of grid map conversion an aerial image shown in figure xxx.

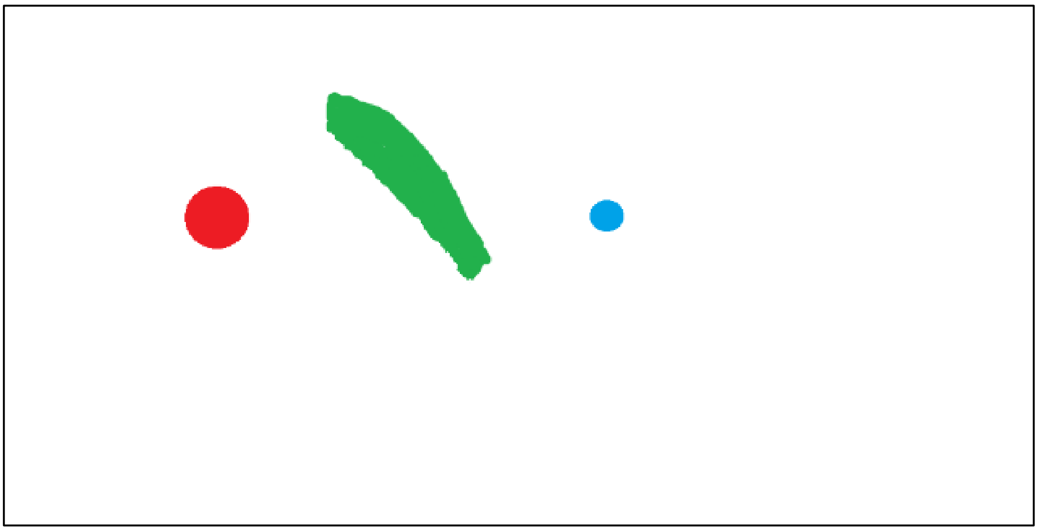

Figure3.3: Sample aerial Image where the RED dot is the ROVER, GREEN region are OBSTACLE and BLUE dot is the TARGET.

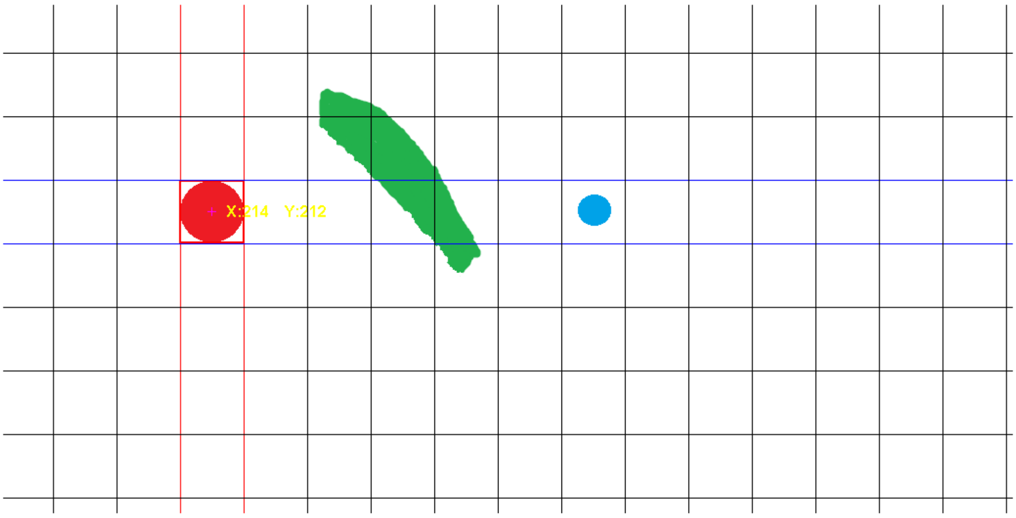

Figure 3.4: Grid map conversion of the image shown in fig 3.3.

The grid produced has 7 rows and 15 columns while counting only the full squares. Hence, a 3D grid map of 7 planes with 7 rows, 15 columns will be produced. The number of rows and columns produced vary with respect to the rover size.

3.2 A star Search algorithm

The star implementation is carried out after the grid map containing node details is created. The searching starts from the start node until it reaches the target node. Once the target node is found out, the step which will take the rover from start to end is calculated. The rover then follows those steps to reach the destination.

It can be divided into the following categories:

- Starting the search: After the map is generated the search begins from the rover position (starting node). The starting point is added to the OPEN list. The Open list is a 1D array and contains the nodes that may fall along the path of traversal or may not. Search is done towards all the walkable nodes ignoring the wall/obstacle and then these neighbour are added to the OPEN list. For each of these neighbour the central node from which searching is done is set as parent.

- Path scoring: Then the F cost is calculated for each of the neighbour.

  $$F = G + H$$

  G = movement cost from the starting node to current exploring node.
  H = movement cost from current exploring node to the target node.

- Continuing the search: To continue the search operation, the node with the lowest F cost movement is chosen from the OPEN list and set as central node & added to the CLOSED list. From this new central node, neighbouring search begins. The neighbour nodes except those on CLOSED LIST or unwalkable are added to the OPEN list there by assigning the central node as parent node. Also, if the adjacent searched neighbour node is already in OPEN then it is check whether the path to that node is better one or not. In other words, it is checked to see whether the G score for that square is lower if we use the current square to get there. If not, don't do anything. If yes, then the G cost of the new path is changed to the lowest value, changing the parent of the adjacent node to the selected central node.

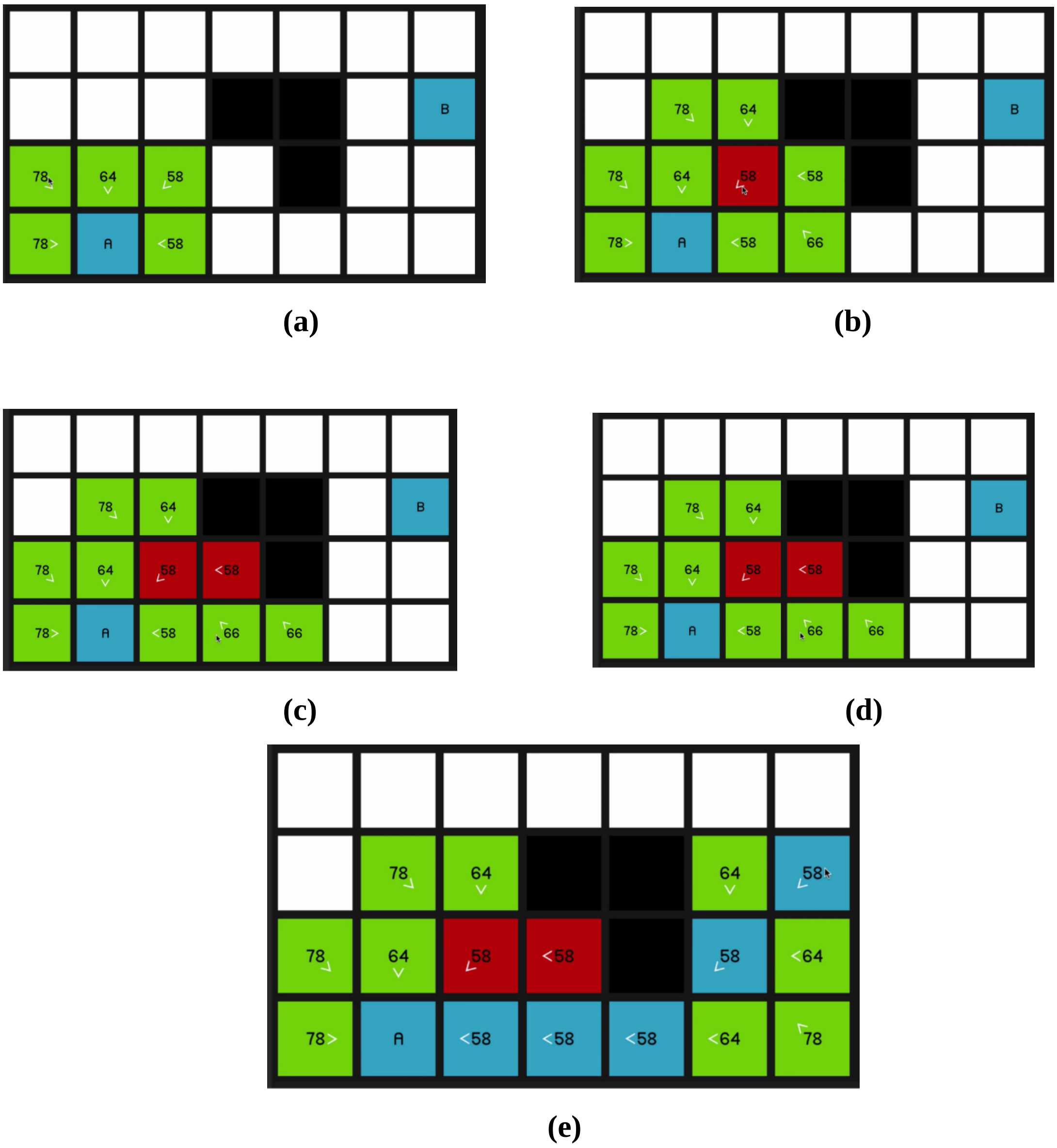

Figure 3.5: (a), (b), (c), (d) & (e) shows the working of the A algorithm step by step.*

3.3 Serial transmission

The transfer of results is done by establishing serial communication between the arduino board on the rover and the matlab computing environment. At first a BAUD rate is set i.e.

9600 and then the control or traversal commands for the rover is sent from the MATLAB to the arduino. Arduino is an open source prototyping board and has the ability to communicate serially with any computer. The arduino is connected via an USB cable to the computer.

Chapter 4 - EXPERIMENTAL SETUP

Arduino Board

Ultrasonic sensor

Castor wheel

Bore wheel

An experimental model of the system is setup in lab. The system comprises of an aerial camera, a computer to process the image and a rover. The aerial camera is a web camera fitted into the celling to take the aerial map of both the bot and the target. The computer is used is Dell Inspiron laptop to process the image taken through the celling camera. The rover consist of arduino Uno micro-controller to process the serial data given by the computer (MATLAB). It also have ultrasonic sensor for local area obstacle detection.

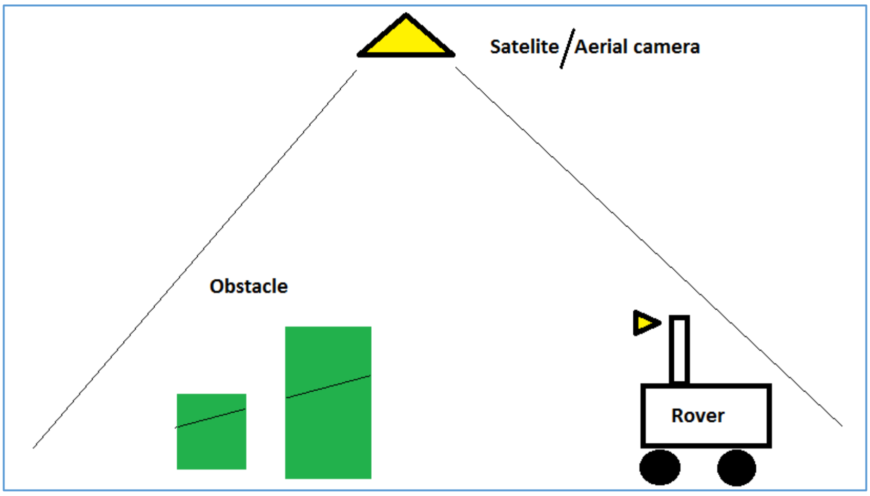

Figure 4.1: The aerial image provided from the satellite to the computer. The aerial image contains both the rover and the obstacle.

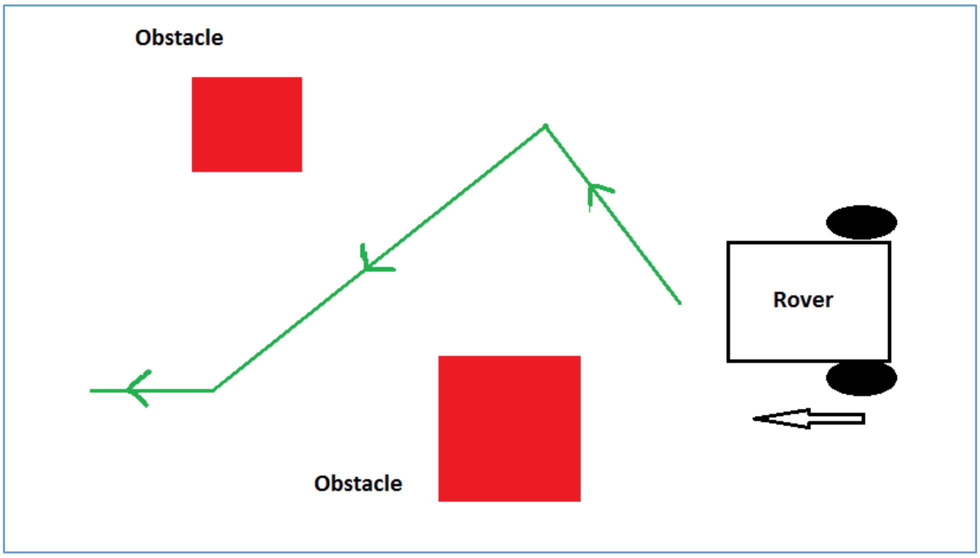

Figure 5.2: Top view of aerial image of image shown in figure 3.

Table 4.1 Rover specification

Microcontroller	Arduino Uno ATmega328
Flash Memory	32 KB (ATmega328)
Operating voltage	5V
SRAM	2KB(ATmega328)
Input Voltage(recommended)	6-12V
Input Voltage (limits)	6-12V
Digital I/O Pins	14(of which 6 provide PWM output)
ANALOG Input Pins	6
Motors	2 Direct Current, 60 RPM DC Motor
Motor Driver	L298, Up to 46V, 2A Dual DC Motor Driver
Speed	Max: 60 RPM, MIN: 30RPM
Sensors	3 Ultrasonic Range Finder Sensor Distance measuring range: 2cm to 400cm
Communication	USB connection Serial Port
Size	Height: 12cm, Length: 17cm, Width: 20cm
Weight	Approx.: 1.2kg
Payload	Approx. 400g
Power	Rechargeable Lithium Polymer 6cell, 11.1V, 2000mAh
Aerial Camera	Web cam

4.1 Arduino Uno

4.1.1 Working

It is a Single-board microcontroller consisting of Atmel 328 AVR chip. It consist of 14 digital I/O pins (in which six of them are PWM pins), 6analog input pins, a 16 MHz ceramic resonator, a USB connection, PWM outputs, a reset button, an ICSP header and a power jack. The rover is controlled using Arduino Uno microcontroller. The inputs from the ultrasonic sensor gives the information about local obstacle to control and command the rover direction.

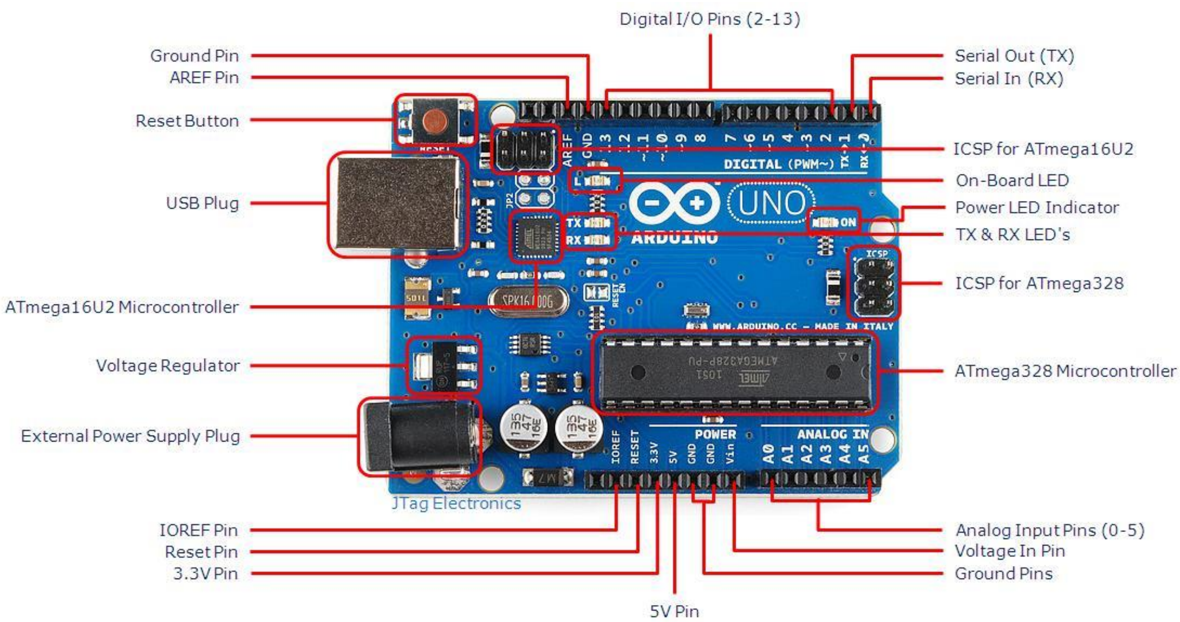

Figure 4.3: Arduino Uno microcontroller.

Table 4.2 Specification of Arduino Uno

Operational Voltage	5V
Input Voltage (recommended)	7-12V
DC Current per I/O pin	20mA
DC Current per 3.3V pin	50mA
Clock Speed	16MHz
Weight	25g

4.2 Ultrasonic sensor

4.2.1 Working

It is a distance measuring device based on the principle of time of flight, Doppler Effect, and the attenuation of sound. It emits small large frequency pulse signal at certain customary interval. The waves travel at the speed of sound. When an obstacle comes in the way of the wave it bounces back to the sensor as echo. The receiver then calculates the time taken by the signal to return back and with the velocity of speed known, it calculate the distance travel. Hence, the half of distance travelled is the distance at which the obstacle is placed.

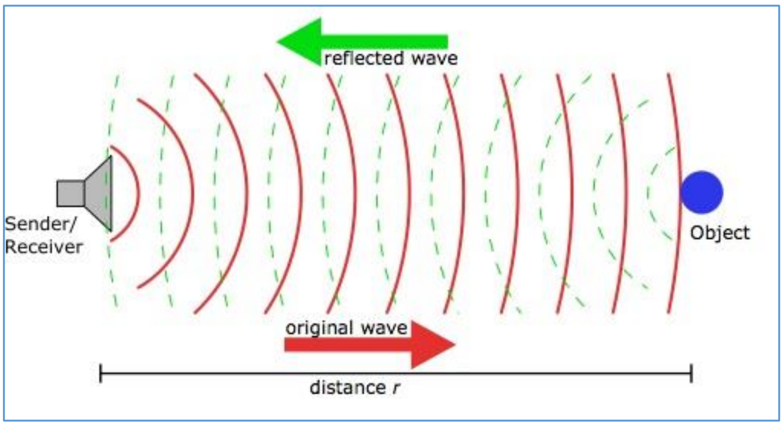

Figure 4.4: Shows how the ultrasonic waves send from the sender is reflected back to the receiver of sensor.

Figure 4.5: Ultrasonic Sensor.

Table 4.3 Specification of Ultrasonic sensor

Working voltage	5V
Working current	15mA
Working frequency	40Hz
Range	Max: 400cm, Min: 2cm
Measuring angle	15 degree
Trigger input Signal	10 icro seconds TTL pulse

4.3 Caster wheel

Ball caster wheel is an Omni directional wheel. It is the neutral wheel which help in supporting the weight of the chassis. It is designed to be mounted on the base of big objects for easy traversal of the object. It is used mostly in smooth environments and flat surfaces.

Figure 4.6: Ball caster wheel for robots.

4.4 Bore wheel

It is a medium sized wheel commonly used for medium duty application. It can coupled with any motor of shaft diameter 6mm. In this project we are using two bore wheel operated by two dc gear motor.

Figure 4.7: Bore wheel.

Table 4.4 Specification of bore wheel

Wheel diameter	76 mm
Wheel thickness	20 mm
Hole diameter	6 mm
Weight	85 gram

Chapter 5 –
CODE IMPLEMENTATION AND RESULTS

Code implementation

Simulation and result

5.1 CODE IMPLEMENTATION

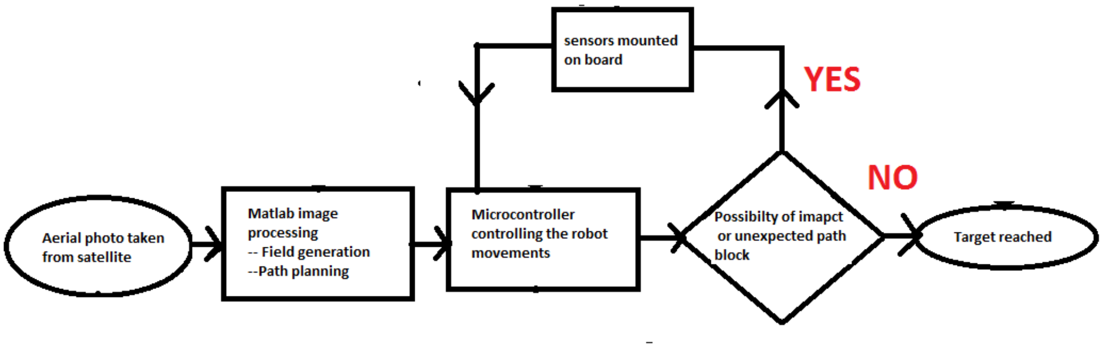

Figure 6.1: Flow diagram of the model.

The first step of path planning is map generation. So, we first read a map then detect the rover, detect the obstacle.

The following algorithm is used for map generation:

- The aerial image is taken through the camera and set to DATA variable.
- The image is then converted to GRAY image following with a filtration process.
- The filtration process includes subtraction of red colour value (here rover is RED coloured), there after it filtered for any unwanted noise
- Now after conversion the result image into binary image we get our rover position on the image.
- Using the BoundingBox tool the width of the rover with respect to image is found out and a grid map is generated with each square cell having size equal to width of the rover. The grid map generated is a 3D array with each cell to store the node information like walkable or un-walkable, pixel coordinates, path cost.
- For obstacle detection each node of the grid is searched obstacle (here obstacle is GREEN coloured) and if it is found the respective node is ladled as unwalkable path.

Figure 5.2: Test image.

Figure 5.3: Binary image showing the location of the rover.

After the grid map is created a window pops up with the original map. Upon selecting the destination the pixel position of the destination is found out using the extracting the mouse button clicking position in the image. From the pixel position the node in which destination is found out.

The path calculation/ path planning is done by implementing the A* algorithm on the grid map generated. The following is the algorithm followed for A* implementation:

- An 1D array name OPEN list is created
- Another 1D array named CLOSED list is created
- At first the start node(rover position) is added to the OPEN list

- Now an infinite Loop is created within which main code is evaluated
    - The node with the lowest f value is set to the current exploring node
    - Then the current node is removed from the OPEN list and
    - added to the CLOSED list
    - If the Current node is the Target node

      Then the execution is stopped and back tracing for path finding
        - For each Neighbour that is explored

          It is checked whether Neighbour is not traversable or is in CLOSED

          If yes then Skip to next Neighbour

          Also new path to each explored Neighbour is checked

          If yes, then the f_value of the Neighbour is changed to new

          And the parent of Neighbour node is set to Current node

          If Neighbour is not in OPEN

          Then it is added to the Neighbour

          Also the f value to new neighbour is calculated
- End loop

5.2 Simulation and results

In result, we are considering some simulation IN MATLAB based on which a mobile robot will traverse. The signal processing and a star code execution both was carried out in MATLAB and subsequent results were noted down. The build mobile robot have three ultrasonic sensors front, left and right. At first the path is generated by path finding system built in the MATALB software. For image processing we are using a digital map in which the RED dot is the rover, GREEN regions are the obstacles and BLUE dot is the destination which is selected using mouse. Here, for our experiment we are considering three different situation and seeing how the system generates path for the mobile robot.

(a) Single obstacle avoidance path

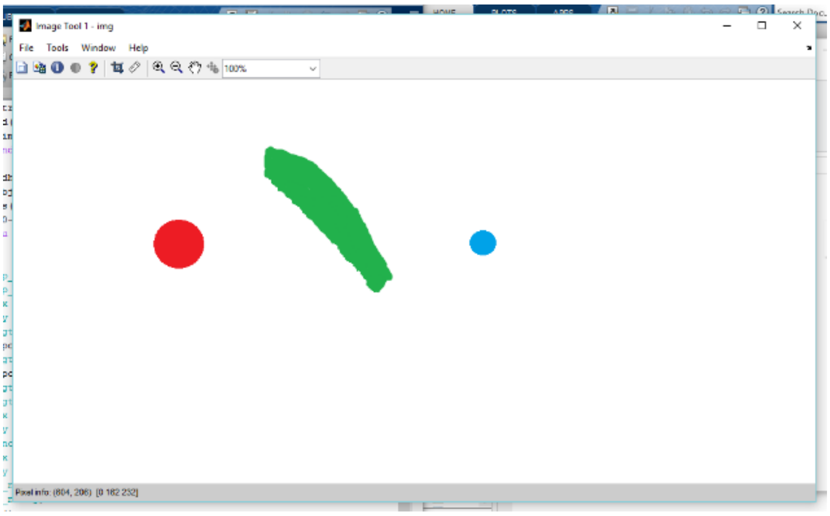

Figure 5.4: Map with single obstacle.

The image here has a rover size comparable with the map and only one obstacle. For grid map generation it took 6 mins to process the image.

After the image is processed i.e. rover detection is done, obstacle detection is done, and grid map is created, the system shows the grid map asking the user to select the target node (here we are taking BLUE dot as target).

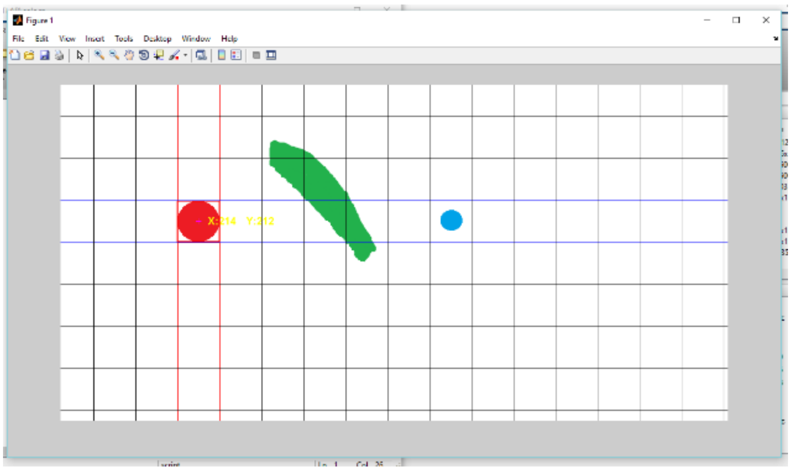

Figure 5.5: Grid map generated pop up window asking for target selection.

Upon selection of target node (BLUE dot) the system took 0.2 seconds to calculate the shortest route for the rover.

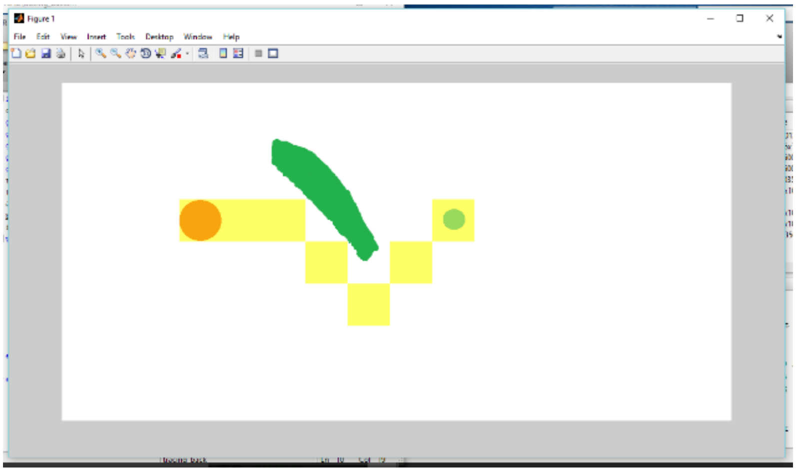

Figure 5.6: Shows path traced between rover and destination.

Table 5.1: Commands generated for environment 1 by the system for the mobile robot

Forward 2 units	Right diagonal Turn	Forward 2 units	Left turn	Forward 2 units

(b) Small rover size with multiple obstacle

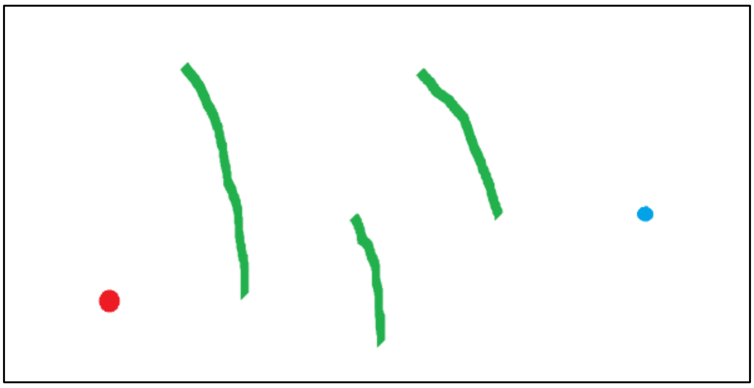

Figure 5.7: map with multiple avoidance.

IN this case the rover is very small as compared with the map size. Hence, the time taken for grid map creation of this image is 9 mins.

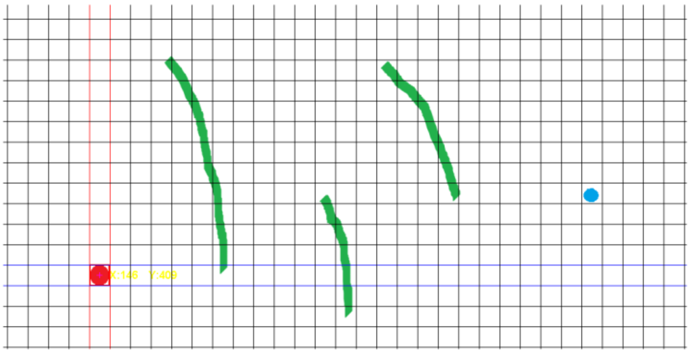

Figure 5.8: grid map of multiple obstacle image.

After the grid map is generated the destination is selected with in the image. And the time taken for the compilation of shortest route algorithm is 0.5 seconds.

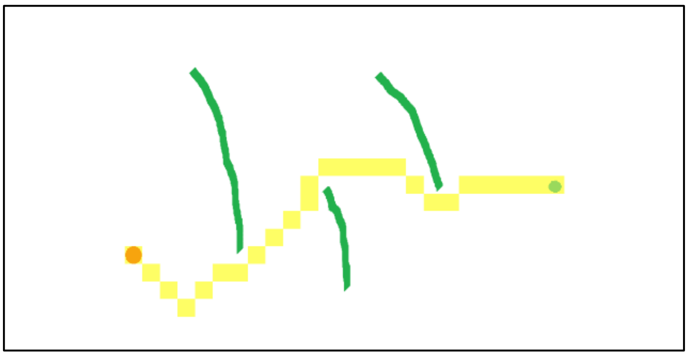

Figure 5.9: Path traced in multiple obstacle map.

Table 5.2: Commands generated for environment 2 by the system for the mobile robot

Right diagonal turn	Forward 3 units	Left turn	Forward 2 units	Right diagonal turn	Forward 1 unit	Left diagonal turn	Forward 4 units

Left diagonal turn	Forward 1 units	Right diagonal turn	Forward 4 units	Right diagonal turn	Forward 3 unit	Left diagonal turn	Forward 1 units

Right diagonal turn	Forward 1 units	Right diagonal turn	Forward 4 units

(c) A map with medium rover size and complex obstacle

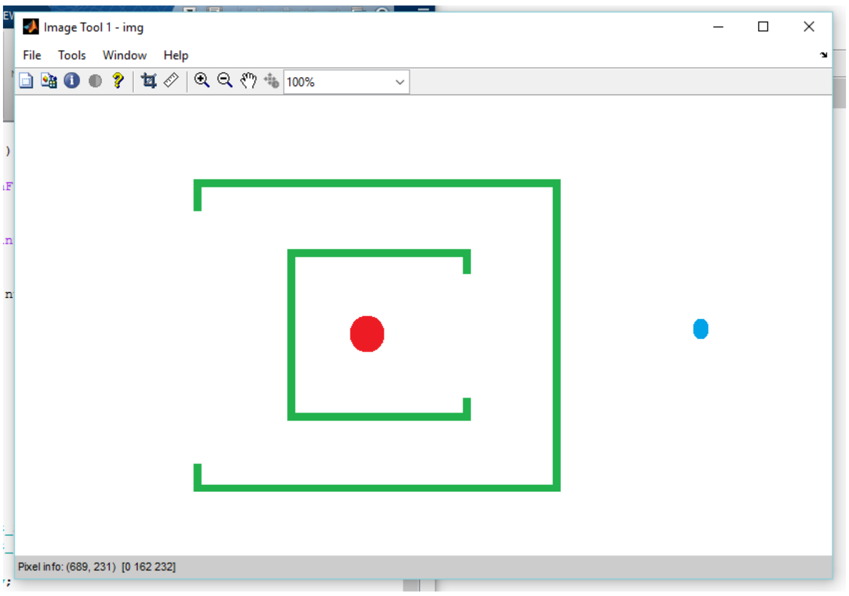

Figure 5.10: map with Medium rover size and clustered environment.

In this image the rover size is not very large but comparable, the obstacle is clustered type. For the grid map generation the processing time was 8 mins. And the time taken for path finding algorithm is 0.4 seconds.

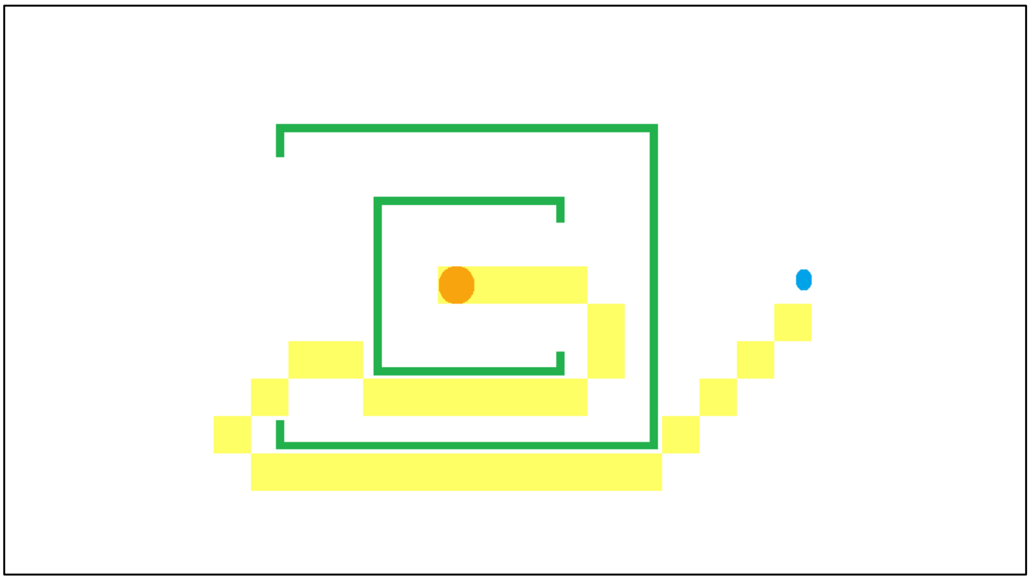

Figure 5.11: Path traced in a clustered environment.

Table 5.3: Commands generated for environment 3 by the system for the mobile robot

Forward 3 units	Right diagonal turn	Right diagonal turn	Forward 2 units	Right diagonal turn	Forward 1 unit	Right diagonal turn	Forward 4 units

Right diagonal turn	Forward 1 unit	Left diagonal turn	Forward 1 units	Left diagonal turn	Forward 2 unit	Left turn	Forward 1 units

Left diagonal turn	Forward 9 unit	Left diagonal turn	Forward 4 units

Table 5.3: Comparison of compilation time for different map environment

Sl. No.	Environment	Image processing time(mins)	Path finding time(seconds)
1	Rover size is comparable to map	6	0.2
2	Rover size is small compared to map	9	0.5
3	Rover size is medium and full of obstacles.	8	0.37(approx. 4)

From the table, by comparing environment 1 & 2 it is find out that the image processing time directly depends on the rover size in the map while it remains independent of obstacle concentration as seen by comparing environment 2 & 3 . With decrease in the rover size the image processing time increases exponentially. In the other hand, the processing time for path finding algorithm is a linearly dependent with the rover size.

Chapter 6 - CONCUSION AND FUTURE SCOPE OF WORK

6.1 Conclusion

Even if there is signal cut-off from the satellite the rover can continue their mission with the help of an autonomous path planning system. In this current research, we have considered different digital maps with different configuration and implemented the proposed path finding algorithm in it. The result was that the system was able to process the image and perform path finding algorithm on it. The success of the robot navigation will depends on the quality of the map image obtained from the satellite. The compilation time and the image processing time depends on the rover size in the map. The design of the system was a successful and the code was successfully executed. In the research different environments are considered to find out the navigation path the rover. In this, we used A star algorithm for navigation of autonomous rover using an aerial map. The simulation result showed that the proposed method enables to reach the destination safely without colliding with the obstacles.

6.2 Scope of future work:

- In this project work the image processing technique used were are the basic object detection and filtration process. Also, the digital images used are low quality. However, with other higher image processing technique and better picture quality the accuracy of this system can be improved.
- Also, with use of other range finding device on board a 3D terrain map can be generated which will significantly increase the path finding method.

REFERENCES

[1] Ono, Masahiro, et al. "Risk-aware planetary rover operation: Autonomous terrain classification and path planning." Aerospace Conference, 2015 IEEE. IEEE, 2015.

[2] Fraundorfer, Friedrich, et al. "Vision-based autonomous mapping and exploration using a quadrotor MAV." Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on. IEEE, 2012.

[3] Qu, Yao-hong, Quan Pan, and Jian-guo Yan. "Flight path planning of UAV based on heuristically search and genetic algorithms." Industrial Electronics Society, 2005. IECON 2005. 31st Annual Conference of IEEE. IEEE, 2005.

[4] Warren, Charles W. "Fast path planning using modified A* method." Robotics and Automation, 1993. Proceedings, 1993 IEEE International Conference on. IEEE, 1993.

[5] Maohai, Li, et al. "Robust omnidirectional mobile robot topological navigation system using omnidirectional vision." Engineering applications of artificial intelligence 26.8 (2013): 1942-1952.

[6] Schmid, Korbinian, et al. "Stereo vision based indoor/outdoor navigation for flying robots." Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on. IEEE, 2013.

[7] Howard, Thomas M., et al. "Enabling continuous planetary rover navigation through FPGA stereo and visual odometry." Aerospace Conference, 2012 IEEE. IEEE, 2012.

[8] Sarkar, Anirban, Sanjeev Srivastava, and B. S. Manoj. "Elevation mapping using stereo vision enabled heterogenous multi-agent robotic network."Global Humanitarian Technology Conference: South Asia Satellite (GHTC-SAS), 2013 IEEE. IEEE, 2013.

[9] Ishigami, Genya, Masatsugu Otsuki, and Takashi Kubota. "Path planning and navigation framework for a planetary exploration rover using a laser range finder." Field and Service Robotics. Springer Berlin Heidelberg, 2014.

[10] Barraquand, Jerome, Bruno Langlois, and Jean-Claude Latombe. "Numerical potential field techniques for robot path planning." Systems, Man and Cybernetics, IEEE Transactions on 22.2 (1992): 224-241.

[11] Kimmel, Ron, and James A. Sethian. "Optimal algorithm for shape from shading and path planning." Journal of Mathematical Imaging and Vision 14.3 (2001): 237-244.

[12] Mezouar, Youcef, and François Chaumette. "Path planning for robust image-based control." Robotics and Automation, IEEE Transactions On 18.4 (2002): 534-549.

[13] Corke, Peter, Dennis Strelow, and Sanjiv Singh. "Omnidirectional visual odometry for a planetary rover." Intelligent Robots and Systems, 2004.(IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on. Vol. 4. IEEE, 2004.

[14] Krogh, Bruce H., and Charles E. Thorpe. "Integrated path planning and dynamic steering control for autonomous vehicles." Robotics and Automation. Proceedings. 1986 IEEE International Conference on. Vol. 3. IEEE, 1986.

[15] Kanayama, Yutaka, and Bruce I. Hartman. "Smooth local path planning for autonomous vehicles." Robotics and Automation, 1989. Proceedings., 1989 IEEE International Conference on. IEEE, 1989.

[16] Hofner, Christian, and Gunther Schmidt. "Path planning and guidance techniques for an autonomous mobile cleaning robot." Intelligent Robots and Systems' 94.'Advanced Robotic Systems and the Real World', IROS'94. Proceedings of the IEEE/RSJ/GI International Conference on. Vol. 1. IEEE, 1994.

[17] Garcia, MA Porta, et al. "Path planning for autonomous mobile robot navigation with ant colony optimization and fuzzy cost function evaluation." Applied Soft Computing 9.3 (2009): 1102-1110.

[18] Garau, Bartolome, Alberto Alvarez, and Gabriel Oliver. "Path planning of autonomous underwater vehicles in current fields with complex spatial variability: an A* approach." Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on. IEEE, 2005.

[19] Yao, Junfeng, et al. "Path planning for virtual human motion using improved A* star algorithm." Information Technology: New Generations (ITNG), 2010 Seventh International Conference on. IEEE, 2010.

[20] Bell, Michael GH. "Hyperstar: A multi-path Astar algorithm for risk averse vehicle navigation." Transportation Research Part B: Methodological 43.1 (2009): 97-107.

[21] Sathyaraj, B. Moses, et al. "Multiple UAVs path planning algorithms: a comparative study." Fuzzy Optimization and Decision Making 7.3 (2008): 257-267.

[22] Seet, Boon-Chong, et al. "A-STAR: A mobile ad hoc routing strategy for metropolis vehicular communications." Networking 2004. Springer Berlin Heidelberg, 2004.

[23] Idrissi, Abdellah, Chu Min Li, and Jean Frédéric Myoupo. "An Algorithm for a Constraint Optimization Problem in Mobile Ad-hoc Networks." Tools with Artificial Intelligence, 2006. ICTAI'06. 18th IEEE International Conference on. IEEE, 2006.

[24] Kwon, Young-Kwan. "A path generation method for an autonomous mobile robot based on a virtual elastic force." The Journal of the Korea institute of electronic communication sciences 8.1 (2013): 149-157.

[25] Stout, Bryan. "Smart moves: Intelligent pathfinding." Game developer magazine 10 (1996): 28-35.

[26] Hwang, Yong K., and Narendra Ahuja. "A potential field approach to path planning." Robotics and Automation, IEEE Transactions on 8.1 (1992): 23-32.

[27] Gennery, Donald B. "Traversability analysis and path planning for a planetary rover." Autonomous Robots 6.2 (1999): 131-146.

[28] Moravec, Hans P. Obstacle avoidance and navigation in the real world by a seeing robot rover. No. STAN-CS-80-813. STANFORD UNIV CA DEPT OF COMPUTER SCIENCE, 1980.

[29] Shum, Alex, Kirsten Morris, and Amir Khajepour. "Direction-dependent optimal path planning for autonomous vehicles." Robotics and Autonomous Systems 70 (2015): 202-214.